_$2

Sym

ALL

ASC

BOD
BOD
BOD
BOD
BOD
BOD
BOD
BOD
BUG
BYP
CAN
CAN
CAN
CHE
CHE

CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU

```
  OOOOOO    PPPPPPPP      CCCCCCCC    OOOOOO    MM      MM  RRRRRRRR      QQQQQQ      SSSSSSSS  TTTTTTTTTT
  OOOOOO    PPPPPPPP      CCCCCCCC    OOOOOO    MM      MM  RRRRRRRR      QQQQQQ      SSSSSSSS  TTTTTTTTTT
OO      OO  PP      PP  CC          OO      OO  MMMM  MMMM  RR      RR  QQ      QQ  SS              TT
OO      OO  PP      PP  CC          OO      OO  MMMM  MMMM  RR      RR  QQ      QQ  SS              TT
OO      OO  PP      PP  CC          OO      OO  MM  MM  MM  RR      RR  QQ      QQ  SS              TT
OO      OO  PP      PP  CC          OO      OO  MM  MM  MM  RR      RR  QQ      QQ  SS              TT
OO      OO  PPPPPPPP    CC          OO      OO  MM      MM  RRRRRRRR    QQ      QQ    SSSSSS        TT
OO      OO  PPPPPPPP    CC          OO      OO  MM      MM  RRRRRRRR    QQ      QQ    SSSSSS        TT
OO      OO  PP          CC          OO      OO  MM      MM  RR   RR     QQ  QQ  QQ          SS      TT
OO      OO  PP          CC          OO      OO  MM      MM  RR   RR     QQ  QQ  QQ          SS      TT
OO      OO  PP          CC          OO      OO  MM      MM  RR    RR    QQ    QQ            SS      TT          ....
OO      OO  PP          CC          OO      OO  MM      MM  RR    RR    QQ    QQ            SS      TT          ....
  OOOOOO    PP            CCCCCCCC    OOOOOO    MM      MM  RR      RR    QQQQ  QQ  SSSSSSSS      TT          ....
  OOOOOO    PP            CCCCCCCC    OOOOOO    MM      MM  RR      RR    QQQQ  QQ  SSSSSSSS      TT          ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE  OPC$OPCOMRQST   (
;     2          0002  0                         LANGUAGE (BLISS32),
;     3          0003  0                         IDENT = 'V04-000'
;     4          0004  0                        ) =
;     5          0005  0
;     6          0006  0 !*****************************************************************************
;     7          0007  0 !*                                                                           *
;     8          0008  0 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;     9          0009  0 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;    10          0010  0 !*  ALL RIGHTS RESERVED.                                                     *
;    11          0011  0 !*                                                                           *
;    12          0012  0 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  0 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  0 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  0 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  0 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  0 !*  TRANSFERRED.                                                             *
;    18          0018  0 !*                                                                           *
;    19          0019  0 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  0 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  0 !*  CORPORATION.                                                             *
;    22          0022  0 !*                                                                           *
;    23          0023  0 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  0 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    25          0025  0 !*                                                                           *
;    26          0026  0 !*                                                                           *
;    27          0027  0 !*****************************************************************************
;    28          0028  0
;    29          0029  0 !++
;    30          0030  0 ! FACILITY:
;    31          0031  0 !
;    32          0032  0 !       OPCOM
;    33          0033  0 !
;    34          0034  0 ! ABSTRACT:
;    35          0035  0 !
;    36          0036  0 !       This module contains the specialized logic to service
;    37          0037  0 !       a particular type of request sent by a user to OPCOM.
;    38          0038  0 !
;    39          0039  0 ! Environment:
;    40          0040  0 !
;    41          0041  0 !       VAX/VMS operating system.
;    42          0042  0 !
;    43          0043  0 ! Author:
;    44          0044  0 !
;    45          0045  0 !       Steven T. Jeffreys
;    46          0046  0 !
;    47          0047  0 ! Creation date:
;    48          0048  0 !
;    49          0049  0 !       March 10, 1981
;    50          0050  0 !
;    51          0051  0 ! Revision history:
;    52          0052  0 !
;    53          0053  0 !       V03-002 CWH3001         CW Hobbs                16-Sep-1983
;    54          0054  0 !               Use jacket routines for VM calls.
;    55          0055  0 !
;    56          0056  0 !       V03-001 CWH3001         CW Hobbs                30-Jul-1983
;    57          0057  0 !               Various and sundry things to make OPCOM distributed
```

```
;    58          0058  0 !                   across the cluster.
;    59          0059  0 !
;    60          0060  0 ! V02-0164      STJ0164         Steven T. Jeffreys,     08-Feb-1982
;    61          0061  0 !                   Make references to library routines use general addressing mode.
;    62          0062  0 !
;    63          0063  0 !--
;    64          0064  0
;    65          0065  1 BEGIN                                           ! Start of OPCOMRQST
;    66          0066  1
;    67          0067  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    68          0068  1 LIBRARY 'LIB$:OPCOMLIB';
;    69          0069  1
;    70          0070  1 FORWARD ROUTINE
;    71          0071  1         REQUEST_HANDLER : NOVALUE,              ! Handle normal request
;    72          0072  1         REQUEST_CLM_HANDLER : NOVALUE,          ! Handle request from remote node
;    73          0073  1         REQUEST_CLM_CHECK_HANDLER : NOVALUE;    ! Check to see if request is in database
;    74          0074  1
;    75          0075  1 BUILTIN
;    76          0076  1         INSQUE,                                 ! Insert entry onto a queue
;    77          0077  1         REMQUE;                                 ! Remove entry from a queue
;    78          0078  1
;    79          0079  1 EXTERNAL
;    80          0080  1         LCL_NOD         : $ref_bblock,
;    81          0081  1         NOD_HEAD        : VECTOR [2, LONG],
;    82          0082  1         GLOBAL_STATUS   : BITVECTOR,            ! Global status bits
;    83          0083  1         REQUEST_NUMBER  : LONG;                 ! Current request #
;    84          0084  1
;    85          0085  1 EXTERNAL ROUTINE
;    86          0086  1         CHECK_REQUEST,                          ! Common sanity checks
;    87          0087  1         CLUSMSG_CONV_CLM_RQCB,                  ! Convert cluster message to RQCB
;    88          0088  1         CLUSMSG_RQCB_SEND,                      ! Send RQCB to remote nodes
;    89          0089  1         CLUSUTIL_INCR_SEQUENCE,                 ! Cluster-unique sequence incrementor
;    90          0090  1         DEALLOCATE_MCB  : NOVALUE,              ! Dispose of an MCB
;    91          0091  1         DEALLOCATE_RQCB : NOVALUE,              ! Dispose of an RQCB
;    92          0092  1         DUMP_LOG_FILE,                          ! Write random string to log file
;    93          0093  1         FIND_OCD,                               ! Find a given OCD
;    94          0094  1         FORMAT_MESSAGE,                         ! Format a message and create an MCB
;    95          0095  1         LOG_MESSAGE,                            ! Write a message to a logfile
;    96          0096  1         NOTIFY_LISTED_OPERATORS,                ! Notify operators on an operator list
;    97          0097  1         SEND_REPLY,                             ! Send reply to a requestor
;    98          0098  1         TRIM_LENGTH;                            ! Length with trailing spaces trimmed
```

```
;  100     0099  1 GLOBAL ROUTINE REQUEST_HANDLER (BUFFER_DESC) : NOVALUE =
;  101     0100  1
;  102     0101  1 !++
;  103     0102  1 ! Functional description:
;  104     0103  1 !
;  105     0104  1 !       This routine is the handler for all REQUEST messages received by OPCOM.
;  106     0105  1 !
;  107     0106  1 !
;  108     0107  1 ! Input:
;  109     0108  1 !
;  110     0109  1 !       BUFFER_DESC : The address of a quadword buffer descriptor that
;  111     0110  1 !                     describes the buffer containing the message.
;  112     0111  1 !
;  113     0112  1 ! Implicit Input:
;  114     0113  1 !
;  115     0114  1 !       None.
;  116     0115  1 !
;  117     0116  1 ! Output:
;  118     0117  1 !
;  119     0118  1 !       None.
;  120     0119  1 !
;  121     0120  1 ! Implict output:
;  122     0121  1 !
;  123     0122  1 !       Some accounting data will be updated
;  124     0123  1 !       to reflect the receipt of the message.
;  125     0124  1 !
;  126     0125  1 ! Side effects:
;  127     0126  1 !
;  128     0127  1 !       None.
;  129     0128  1 !
;  130     0129  1 ! Routine value:
;  131     0130  1 !
;  132     0131  1 !       None.
;  133     0132  1 !--
;  134     0133  1
;  135     0134  2 BEGIN                                           ! Start of REQUEST_HANDLER
;  136     0135  2
;  137     0136  2 MAP
;  138     0137  2
;  139     0138  2         BUFFER_DESC     : $ref_bblock;
;  140     0139  2
;  141     0140  2 LOCAL
;  142     0141  2         MESSAGE_VECTOR  : VECTOR [9, LONG],     ! Message info
;  143     0142  2         ON_BUF          : VECTOR [64, BYTE],    ! Buffer for preposition (" on " node)
;  144     0143  2         ON_DSC          : VECTOR [2, LONG]      ! Desc for preposition
;  145     0144  2                         INITIAL (64, ON_BUF),
;  146     0145  2         RQCB            : $ref_bblock,          ! RQCB data structure
;  147     0146  2         OCD             : $ref_bblock,          ! OCD data structure
;  148     0147  2         MCB             : $ref_bblock,          ! MCB data structure
;  149     0148  2         MSG             : $ref_bblock,          ! Pointer to user request
;  150     0149  2         FOUND           : LONG,                 ! Boolean
;  151     0150  2         SCOPE           : LONG,                 ! Scope of request
;  152     0151  2         SCOPE_LIMIT     : LONG,                 ! Loop control
;  153     0152  2         STATUS          : LONG;
;  154     0153  2
;  155     0154  2 EXTERNAL
;  156     0155  2         LCL_NODENAME    : $bblock;              ! Name of local node (DECnet or cluster)
```

```
;  157          0156  2 !
;  158          0157  2 ! Make sure there is enough data in the request.
;  159          0158  2 !
;  160          0159  2 IF .BUFFER_DESC [DSC$W_LENGTH] LSS (OPC$K_COMHDRSIZ + OPC$K_REQUEST_MIN_SIZE)
;  161          0160  2 THEN
;  162          0161  2     RETURN;                                         ! Ignore the request
;  163          0162  2 !
;  164          0163  2 ! Do some common sanity checks.
;  165          0164  2 !
;  166          0165  2 IF NOT CHECK_REQUEST (.BUFFER_DESC, RQCB)
;  167          0166  2 THEN
;  168          0167  2     RETURN;
;  169          0168  2 MESSAGE_VECTOR [0] = 0;                           ! Assume no errors
;  170          0169  2 !
;  171          0170  2 ! See if the requestor is issuing this request on another's behalf.
;  172          0171  2 ! If so, and the requestor does not have the privilege to do so,
;  173          0172  2 ! then dismiss the request.
;  174          0173  2 !
;  175          0174  2 IF .RQCB [RQCB_L_SENDERUIC] NEQ .RQCB [RQCB_L_UIC]
;  176          0175  2 THEN
;  177          0176  3     IF  (NOT .$bblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_OPER])
;  178          0177  3     THEN
;  179          0178  3         IF NOT ((.$bblock [RQCB [RQCB_L_SENDERUIC], 2,0,16,0] EQL .$bblock [RQCB [RQCB_L_UIC], 2,0,16,0]) AN
;  180          0179  3                 (.$bblock [RQCB [RQCB_L_PRIVMASK1], PRV$V_GROUP]))
;  181          0180  3         THEN
;  182          0181  3             BEGIN
;  183          0182  3             MESSAGE_VECTOR [0] = OPC$_ILLRQST;
;  184          0183  3             MESSAGE_VECTOR [1] = 0;
;  185          0184  3             END;
;  186          0185  2 !
;  187          0186  2 ! Create a descriptor within the RQCB to point to the request text.
;  188          0187  2 !
;  189          0188  2 MSG = .BUFFER_DESC [DSC$A_POINTER] + OPC$K_COMHDRSIZ;
;  190          0189  2 RQCB [RQCB_L_TEXT_LEN] = .MSG [OPC$W_REQUEST_LENGTH];
;  191          0190  3 IF (.RQCB [RQCB_L_TEXT_LEN] GTR 0)
;  192          0191  2 THEN
;  193          0192  3     BEGIN
;  194          0193  3     !
;  195          0194  3     ! Create a buffer for the request text and copy the text to the buffer.
;  196          0195  3     !
;  197          0196  4     IF NOT (STATUS = OPC$GET_VM (RQCB [RQCB_L_TEXT_LEN], RQCB [RQCB_L_TEXT_PTR]))
;  198          0197  3     THEN
;  199          0198  4         BEGIN
;  200          0199  4         DEALLOCATE_RQCB (.RQCB);
;  201          0200  4         RETURN;
;  202          0201  4         END;
;  203          0202  3     CH$MOVE (.RQCB [RQCB_L_TEXT_LEN], MSG [OPC$T_REQUEST_TEXT], .RQCB [RQCB_L_TEXT_PTR]);
;  204          0203  3     END
;  205          0204  2 ELSE
;  206          0205  3     BEGIN
;  207          0206  3     !
;  208          0207  3     ! There is no request text.  Inform the requestor that this is not allowed.
;  209          0208  3     !
;  210          0209  3     MESSAGE_VECTOR [0] = OPC$_ILLRQST;
;  211          0210  3     MESSAGE_VECTOR [1] = 0;
;  212          0211  2     END;
;  213          0212  2 !
```

```
;  214     0213  2 ! Find an OCD that can handle this request. The OCD is selected
;  215     0214  2 ! according to the SCOPE and UIC of the requestor.  If the SCOPE
;  216     0215  2 ! is unspecified, then look for operator coverage starting in the
;  217     0216  2 ! least privileged scope and continuing to the most privileged.
;  218     0217  2 ! If no OCD is found, then dismiss the request.
;  219     0218  2 !
;  220     0219  2 IF (.RQCB [RQCB_B_SCOPE] EQL OPC$K_UNSPEC)
;  221     0220  2 THEN
;  222     0221  2     SCOPE_LIMIT = OPC$K_SYSTEM
;  223     0222  2 ELSE
;  224     0223  2     SCOPE_LIMIT = .RQCB [RQCB_B_SCOPE];
;  225     0224  2 FOUND = FALSE;
;  226     0225  2 SCOPE = .RQCB [RQCB_B_SCOPE];
;  227     0226  2 WHILE (.SCOPE GEQ .SCOPE_LIMIT) AND (NOT .FOUND) DO
;  228     0227  3     IF NOT (FOUND = FIND_OCD (.SCOPE, .RQCB [RQCB_L_UIC], OCD))
;  229     0228  3     THEN
;  230     0229  3         SCOPE = .SCOPE - 1;
;  231     0230  2 IF NOT .FOUND
;  232     0231  2 THEN
;  233     0232  3     BEGIN
;  234     0233  3     MESSAGE_VECTOR [0] = OPC$_NOPERATOR;                ! No operator coverage
;  235     0234  3     MESSAGE_VECTOR [1] = 0;
;  236     0235  3     END;
;  237     0236  2 !
;  238     0237  2 ! If there is an error message to output,
;  239     0238  2 ! do so and dismiss the request.
;  240     0239  2 !
;  241     0240  2 IF .MESSAGE_VECTOR [0] NEQ 0
;  242     0241  2 THEN
;  243     0242  3     BEGIN
;  244     0243  3     FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
;  245     0244  3     SEND_REPLY (.RQCB, MESSAGE_VECTOR);
;  246     0245  3     DEALLOCATE_RQCB (.RQCB);
;  247     0246  3     RETURN;
;  248     0247  2     END;
;  249     0248  2 !
;  250     0249  2 ! Set the scope of the request.
;  251     0250  2 ! Format the request message and send it to all
;  252     0251  2 ! interested operators on the OCD's operator list.
;  253     0252  2 !
;  254     0253  2 RQCB [RQCB_L_OCD] = .OCD;                           ! Save OCD address
;  255     0254  2 RQCB [RQCB_B_SCOPE] = .OCD [OCD_B_SCOPE];           ! Set request scope
;  256     0255  2 IF .LCL_NODENAME [DSC$W_LENGTH] NEQ 0
;  257     0256  2 THEN
;  258     0257  3     BEGIN
;  259     0258  4     IF NOT (STATUS = $GETMSG (MSGID=OPC$_ON_NODE, MSGLEN=ON_DSC, BUFADR=ON_DSC, FLAGS=1))
;  260     0259  3     THEN
;  261     0260  3         $signal_stop (.STATUS);
;  262     0261  3     END
;  263     0262  2 ELSE
;  264     0263  2     ON_DSC [0] = 0;
;  265     0264  2 IF .RQCB [RQCB_W_REPLYMBX] NEQ 0                    ! Set the message code
;  266     0265  2 THEN
;  267     0266  3     BEGIN                                           ! Request with reply expected
;  268     0267  3     REQUEST_NUMBER = CLUSUTIL_INCR_SEQUENCE (.REQUEST_NUMBER);  ! Increment the number of request
;  269     0268  3     RQCB [RQCB_L_RQSTNUM] = .REQUEST_NUMBER;        ! Set the request number
;  270     0269  3     MESSAGE_VECTOR [0] = OPC$_USERQST;              ! Set the message code
```

```
;  271     0270  3          MESSAGE_VECTOR [1] = 0;                                  ! Set the message Nargs
;  272     0271  3          MESSAGE_VECTOR [2] = .RQCB [RQCB_L_RQSTNUM];             ! Set the request number
;  273     0272  3          MESSAGE_VECTOR [3] = .RQCB [RQCB_W_USERNAMELEN];         ! Set the username string length
;  274     0273  3          MESSAGE_VECTOR [4] = RQCB [RQCB_T_USERNAME];             ! Set the username string addr
;  275     0274  3          MESSAGE_VECTOR [5] = ON_DSC;                             ! The " on " field
;  276     0275  3          MESSAGE_VECTOR [6] = .LCL_NODENAME [DSC$W_LENGTH];       ! Length of nodename
;  277     0276  3          MESSAGE_VECTOR [7] = .LCL_NODENAME [DSC$A_POINTER];      ! Length of nodename
;  278     0277  3          MESSAGE_VECTOR [8] = RQCB [RQCB_L_TEXT_LEN];             ! Set address request descriptor
;  279     0278  3          END
;  280     0279  2      ELSE
;  281     0280  3          BEGIN                                                    ! Request with no reply expected
;  282     0281  3          MESSAGE_VECTOR [0] = OPC$_USERMSG;                       ! Set message code
;  283     0282  3          MESSAGE_VECTOR [1] = 0;                                  ! Set number of paramters
;  284     0283  3          MESSAGE_VECTOR [2] = .RQCB [RQCB_W_USERNAMELEN];         ! Set the username string length
;  285     0284  3          MESSAGE_VECTOR [3] = RQCB [RQCB_T_USERNAME];             ! Set the username string addr
;  286     0285  3          MESSAGE_VECTOR [4] = ON_DSC;                             ! The " on " field
;  287     0286  3          MESSAGE_VECTOR [5] = .LCL_NODENAME [DSC$W_LENGTH];       ! Length of nodename
;  288     0287  3          MESSAGE_VECTOR [6] = .LCL_NODENAME [DSC$A_POINTER];      ! Length of nodename
;  289     0288  3          MESSAGE_VECTOR [7] = RQCB [RQCB_L_TEXT_LEN];             ! Set address request descriptor
;  290     0289  3          END;
;  291     0290  2      FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);                      ! Format the message
;  292     0291  2      IF NOTIFY_LISTED_OPERATORS (.RQCB)
;  293     0292  2      THEN
;  294     0293  3          BEGIN
;  295     0294  3          !
;  296     0295  3          ! At least one operator was notified of the request, so send it off to the cluster.
;  297     0296  3          ! Note that NOTIFY_LISTED_OPERATORS returns true if a remote operator is enabled for the
;  298     0297  3          ! request, even if no operators on the local node were notified.
;  299     0298  3          !
;  300     0299  3          CLUSMSG_RQCB_SEND (-1, CLM__REQUEST, .RQCB);             ! Send it everywhere
;  301     0300  3          !
;  302     0301  3          ! If the request expects a reply, then queue the RQCB
;  303     0302  3          ! onto the OCD for future reference.  Log the request.
;  304     0303  3          !
;  305     0304  3          LOG_MESSAGE (.RQCB);
;  306     0305  3          IF .RQCB [RQCB_W_REPLYMBX] NEQ 0
;  307     0306  3          THEN
;  308     0307  4              BEGIN
;  309     0308  4              INSQUE (.RQCB, .OCD [OCD_L_RQSTFLINK]);
;  310     0309  4              OCD [OCD_W_RQSTCOUNT] = .OCD [OCD_W_RQSTCOUNT] + 1;
;  311     0310  4              $bblock [RQCB [RQCB_L_OPTIONS], OPC$V_NOBRD] = 0;   ! Clear option bits
;  312     0311  4              $bblock [RQCB [RQCB_L_OPTIONS], OPC$V_NOLOG] = 0;
;  313     0312  4              END
;  314     0313  3          ELSE
;  315     0314  3              DEALLOCATE_RQCB (.RQCB);                             ! Dellocate the RQCB
;  316     0315  3          END
;  317     0316  2      ELSE
;  318     0317  3          BEGIN
;  319     0318  3          !
;  320     0319  3          ! None of the operators on the OCD's operator list were
;  321     0320  3          ! enabled to receive the request.  If no reply is expected,
;  322     0321  3          ! just return.  If a reply was expected, then cancel the
;  323     0322  3          ! request and log the cancelation.
;  324     0323  3          !
;  325     0324  3          IF .RQCB [RQCB_W_REPLYMBX] NEQ 0
;  326     0325  3          THEN
;  327     0326  4              BEGIN
```

```
;  328          0327  4          MESSAGE_VECTOR [0] = OPC$_NOPERATOR;
;  329          0328  4          MESSAGE_VECTOR [1] = 0;
;  330          0329  4          FORMAT_MESSAGE (.RQCB, MESSAGE_VECTOR);
;  331          0330  4          SEND_REPLY (.RQCB);
;  332          0331  4          LOG_MESSAGE (.RQCB);
;  333          0332  3          END;
;  334          0333  3      DEALLOCATE_RQCB (.RQCB);
;  335          0334  2      END;
;  336          0335  2
;  337          0336  1 END;                                    ! End of REQUEST_HANDLER

                                                .TITLE  OPC$OPCOMRQST
                                                .IDENT  \V04-000\

                                                .EXTRN  LCL_NOD, NOD_HEAD
                                                .EXTRN  GLOBAL_STATUS, REQUEST_NUMBER
                                                .EXTRN  CHECK_REQUEST, CLUSMSG_CONV_CLM_RQCB
                                                .EXTRN  CLUSMSG_RQCB_SEND
                                                .EXTRN  CLUSUTIL_INCR_SEQUENCE
                                                .EXTRN  DEALLOCATE_MCB, DEALLOCATE_RQCB
                                                .EXTRN  DUMP_LOG_FILE, FIND_OCD
                                                .EXTRN  FORMAT_MESSAGE, LOG_MESSAGE
                                                .EXTRN  NOTIFY_LISTED_OPERATORS
                                                .EXTRN  SEND_REPLY, TRIM_LENGTH
                                                .EXTRN  LCL_NODENAME, OPC$GET_VM
                                                .EXTRN  SYS$GETMSG, LIB$STOP

                                                .PSECT  $CODE$,NOWRT,2

                                  0FFC 00000    .ENTRY  REQUEST_HANDLER, Save R2,R3,R4,R5,R6,R7,R8,-; 0099
                                                        R9,R10,R11
                          5B      0000G  CF 9E 00002    MOVAB   FORMAT_MESSAGE, R11                   ;
                          5A      0000G  CF 9E 00007    MOVAB   LCL_NODENAME, R10                     ;
                          5E        8C   AE 9E 0000C    MOVAB   -116(SP), SP                          ;
                    08    AE        40   8F 9A 00010    MOVZBL  #64, ON_DSC                           ; 0134
                    0C    AE        10   AE 9E 00015    MOVAB   ON_BUF, ON_DSC+4                      ;
                          52        04   AC D0 0001A    MOVL    BUFFER_DESC, R2                       ; 0159
                  0042    8F             62 B1 0001E    CMPW    (R2), #66                             ;
                                         01 1E 00023    BGEQU   1$                                    ;
                                            04 00025    RET                                           ;
                                  4004   8F BB 00026 1$: PUSHR   #^M<R2,SP>                           ; 0165
                 0000G    CF              02 FB 0002A    CALLS   #2, CHECK_REQUEST                     ;
                          01              50 E8 0002F    BLBS    R0, 2$                                ;
                                             04 00032    RET                                           ;
                                    50    AE D4 00033 2$: CLRL    MESSAGE_VECTOR                        ; 0168
                                          6E D0 00036    MOVL    RQCB, R6                              ; 0174
                                    50    38 A6 9E 00039    MOVAB   56(R6), R0                         ;
                                    57    68 A6 9E 0003D    MOVAB   104(R6), R7                        ;
                                    67       60 D1 00041    CMPL    (R0), (R7)                         ;
                                             1B 13 00044    BEQL    4$                                 ;
                    16      32    A6       02 E0 00046    BBS     #2, 50(R6), 4$                     ; 0176
                            02    A7    02 A0 B1 0004B    CMPW    2(R0), 2(R7)                       ; 0178
                                             04 12 00050    BNEQ    3$                                 ;
                          0B        31   A6 E8 00052    BLBS    49(R6), 4$                             ; 0179
                    50    AE 0005807C    8F D0 00056 3$: MOVL    #360572, MESSAGE_VECTOR                ; 0182
                                    54    AE D4 0005E    CLRL    MESSAGE_VECTOR+4                       ; 0183
```

```
                52      04  A2                  26  C1 00061 4$:    ADDL3   #38, 4(R2), MSG                 ; 0188
                                58      0084    C6  9E 00066        MOVAB   132(R6), R8                     ; 0189
                                68        1A    A2  3C 0006B        MOVZWL  26(MSG), (R8)
                                                1A  15 0006F        BLEQ    5$                              ; 0190
                                          0088  C6  9F 00071        PUSHAB  136(R6)                         ; 0196
                                                58  DD 00075        PUSHL   R8
                          0000G CF              02  FB 00077        CALLS   #2, OPC$GET_VM
                                59              50  D0 0007C        MOVL    R0, STATUS
                                69              59  E9 0007F        BLBC    STATUS, 12$
              0088      D6      1C  A2            68  28 00082        MOVC3   (R8), 28(MSG), @136(R6)         ; 0202
                                                0B  11 00089        BRB     6$                              ; 0190
                                50  AE 0005807C  8F  D0 0008B 5$:    MOVL    #360572, MESSAGE_VECTOR         ; 0209
                                          54    AE  D4 00093        CLRL    MESSAGE_VECTOR+4                ; 0210
                                04        53    A6  91 00096 6$:    CMPB    83(R6), #4                      ; 0219
                                                05  12 0009A        BNEQ    7$
                                53              01  D0 0009C        MOVL    #1, SCOPE_LIMIT                 ; 0221
                                                04  11 0009F        BRB     8$
                                53        53    A6  9A 000A1 7$:    MOVZBL  83(R6), SCOPE_LIMIT             ; 0223
                                          50    D4  000A5 8$:       CLRL    FOUND                           ; 0224
                                52        53    A6  9A 000A7        MOVZBL  83(R6), SCOPE                   ; 0225
                                53              52  D1 000AB 9$:    CMPL    SCOPE, SCOPE_LIMIT              ; 0226
                                                16  19 000AE        BLSS    10$
                                21              50  E8 000B0        BLBS    FOUND, 11$
                                          04    AE  9F 000B3        PUSHAB  OCD                             ; 0227
                                                67  DD 000B6        PUSHL   (R7)
                                                52  DD 000B8        PUSHL   SCOPE
                          0000G CF              03  FB 000BA        CALLS   #3, FIND_OCD
                                E9              50  E8 000BF        BLBS    FOUND, 9$
                                                52  D7 000C2        DECL    SCOPE                           ; 0229
                                                E5  11 000C4        BRB     9$                              ; 0227
                                0B              50  E8 000C6 10$:   BLBS    FOUND, 11$                      ; 0230
                50      AE 00058061             8F  D0 000C9        MOVL    #360545, MESSAGE_VECTOR         ; 0233
                                          54    AE  D4 000D1        CLRL    MESSAGE_VECTOR+4                ; 0234
                                          50    AE  D5 000D4 11$:   TSTL    MESSAGE_VECTOR                  ; 0240
                                                15  13 000D7        BEQL    13$
                                          50    AE  9F 000D9        PUSHAB  MESSAGE_VECTOR                  ; 0243
                                                56  DD 000DC        PUSHL   R6
                                6B              02  FB 000DE        CALLS   #2, FORMAT_MESSAGE
                                          50    AE  9F 000E1        PUSHAB  MESSAGE_VECTOR                  ; 0244
                                                56  DD 000E4        PUSHL   R6
              0000G     CF                    02  FB 000E6        CALLS   #2, SEND_REPLY
                                          010D  31  000EB 12$:      BRW     19$                             ; 0245
                                52        04    AE  D0 000EE 13$:   MOVL    OCD, R2                         ; 0253
                          24    A6              52  D0 000F2        MOVL    R2, 36(R6)
                          53    A6        0B    A2  90 000F6        MOVB    11(R2), 83(R6)                  ; 0254
                                                6A  B5 000FB        TSTW    LCL_NODENAME                    ; 0255
                                                26  13 000FD        BEQL    14$
                                7E              01  7D 000FF        MOVQ    #1, -(SP)                       ; 0258
                                          10    AE  9F 00102        PUSHAB  ON_DSC
                                          14    AE  9F 00105        PUSHAB  ON_DSC
                                      000582AB  8F  DD 00108        PUSHL   #361131
              00000000G         00              05  FB 0010E        CALLS   #5, SYS$GETMSG
                                59              50  D0 00115        MOVL    R0, STATUS
                                0D              59  E8 00118        BLBS    STATUS, 15$
                                                59  DD 0011B        PUSHL   STATUS                          ; 0260
              00000000G         00              01  FB 0011D        CALLS   #1, LIB$STOP
                                                04  00124        RET
```

```
                        08   AE   D4 00125 14$:    CLRL    ON_DSC                                          ; 0263
                53      3C   A6   9E 00128 15$:    MOVAB   60(R6), R3                                      ; 0273
                        2E   A6   B5 0012C         TSTW    46(R6)                                          ; 0264
                             41   13 0012F         BEQL    16$
                     0000G   CF   DD 00131         PUSHL   REQUEST_NUMBER                                  ; 0267
          0000G     CF             01 FB 00135    CALLS   #1, CLUSUTIL_INCR_SEQUENCE
          0000G     CF             50 D0 0013A    MOVL    R0, REQUEST_NUMBER
            70      A6      0000G  CF D0 0013F    MOVL    REQUEST_NUMBER, 112(R6)                         ; 0268
            50      AE  000580AB   8F D0 00145    MOVL    #360619, MESSAGE_VECTOR                         ; 0269
                           54      AE D4 0014D    CLRL    MESSAGE_VECTOR+4                                ; 0270
            58      AE      70      A6 D0 00150    MOVL    112(R6), MESSAGE_VECTOR+8                       ; 0271
            5C      AE      74      A6 3C 00155    MOVZWL  116(R6), MESSAGE_VECTOR+12                      ; 0272
            60      AE              53 D0 0015A    MOVL    R3, MESSAGE_VECTOR+16                           ; 0273
            64      AE      08      AE 9E 0015E    MOVAB   ON_DSC, MESSAGE_VECTOR+20                       ; 0274
            68      AE              6A 3C 00163    MOVZWL  LCL_NODENAME, MESSAGE_VECTOR+24                 ; 0275
            6C      AE      04      AA D0 00167    MOVL    LCL_NODENAME+4, MESSAGE_VECTOR+28               ; 0276
            70      AE              58 D0 0016C    MOVL    R8, MESSAGE_VECTOR+32                           ; 0277
                                    26 11 00170    BRB     17$                                             ; 0264
            50      AE  000580B3   8F D0 00172 16$:    MOVL    #360627, MESSAGE_VECTOR                     ; 0281
                           54      AE D4 0017A    CLRL    MESSAGE_VECTOR+4                                ; 0282
            58      AE      74      A6 3C 0017D    MOVZWL  116(R6), MESSAGE_VECTOR+8                       ; 0283
            5C      AE              53 D0 00182    MOVL    R3, MESSAGE_VECTOR+12                           ; 0284
            60      AE      08      AE 9E 00186    MOVAB   ON_DSC, MESSAGE_VECTOR+16                       ; 0285
            64      AE              6A 3C 0018B    MOVZWL  LCL_NODENAME, MESSAGE_VECTOR+20                 ; 0286
            68      AE      04      AA D0 0018F    MOVL    LCL_NODENAME+4, MESSAGE_VECTOR+24               ; 0287
            6C      AE              58 D0 00194    MOVL    R8, MESSAGE_VECTOR+28                           ; 0288
                            50      AE 9F 00198 17$:    PUSHAB  MESSAGE_VECTOR                          ; 0290
                                    56 DD 0019B    PUSHL   R6
                    6B              02 FB 0019D    CALLS   #2, FORMAT_MESSAGE
                                    56 DD 001A0    PUSHL   R6                                              ; 0291
          0000G     CF              01 FB 001A2    CALLS   #1, NOTIFY_LISTED_OPERATORS
                    2B              50 E9 001A7    BLBC    R0, 18$
                                    56 DD 001AA    PUSHL   R6                                              ; 0299
                                    0E DD 001AC    PUSHL   #14
                    7E              01 CE 001AE    MNEGL   #1, -(SP)
          0000G     CF              03 FB 001B1    CALLS   #3, CLUSMSG_RQCB_SEND
                                    56 DD 001B6    PUSHL   R6                                              ; 0304
          0000G     CF              01 FB 001B8    CALLS   #1, LOG_MESSAGE
                            2E      A6 B5 001BD    TSTW    46(R6)                                          ; 0305
                                    39 13 001C0    BEQL    19$
            3C      B2              66 0E 001C2    INSQUE  (R6), @60(R2)                                   ; 0308
                    50      04      AE D0 001C6    MOVL    OCD, R0                                         ; 0309
                            3A      A0 B6 001CA    INCW    58(R0)
                    50              6E D0 001CD    MOVL    RQCB, R0                                        ; 0310
            54      A0              03 8A 001D0    BICB2   #3, 84(R0)                                      ; 0311
                                    04 001D4       RET                                                     ; 0305
                            2E      A6 B5 001D5 18$:    TSTW    46(R6)                                      ; 0324
                                    21 13 001D8    BEQL    19$
            50      AE  00058061   8F D0 001DA     MOVL    #360545, MESSAGE_VECTOR                         ; 0327
                           54      AE D4 001E2    CLRL    MESSAGE_VECTOR+4                                ; 0328
                            50      AE 9F 001E5    PUSHAB  MESSAGE_VECTOR                                  ; 0329
                                    56 DD 001E8    PUSHL   R6
                    6B              02 FB 001EA    CALLS   #2, FORMAT_MESSAGE
                                    56 DD 001ED    PUSHL   R6                                              ; 0330
          0000G     CF              01 FB 001EF    CALLS   #1, SEND_REPLY
                                    56 DD 001F4    PUSHL   R6                                              ; 0331
          0000G     CF              01 FB 001F6    CALLS   #1, LOG_MESSAGE
```

```
                                 56  DD 001FB 19$:    PUSHL   R6                                      ; 0333
                      0000G  CF  01  FB 001FD         CALLS   #1, DEALLOCATE_RQCB                     ;
                                 04 00202             RET                                             ; 0336

; Routine Size:  515 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  339     0337  1 GLOBAL ROUTINE REQUEST_CLM_HANDLER (BUFFER_DESC : $ref_bblock, CLM : $ref_bblock, LEN) : NOVALUE =
;  340     0338  1
;  341     0339  1 !++
;  342     0340  1 ! Functional description:
;  343     0341  1 !
;  344     0342  1 !       This routine is the handler for all REQUEST messages received by OPCOM from remote nodes.
;  345     0343  1 !
;  346     0344  1 !
;  347     0345  1 ! Input:
;  348     0346  1 !
;  349     0347  1 !       BUFFER_DESC -   pointer to message from remote node, including $SNDOPR header
;  350     0348  1 !       CLM -           pointer to CLMRQCB structure
;  351     0349  1 !       LEN -           length of LEN
;  352     0350  1 !
;  353     0351  1 ! Implicit Input:
;  354     0352  1 !
;  355     0353  1 !       None.
;  356     0354  1 !
;  357     0355  1 ! Output:
;  358     0356  1 !
;  359     0357  1 !       None.
;  360     0358  1 !
;  361     0359  1 ! Implict output:
;  362     0360  1 !
;  363     0361  1 !       Some accounting data will be updated
;  364     0362  1 !       to reflect the receipt of the message.
;  365     0363  1 !
;  366     0364  1 ! Side effects:
;  367     0365  1 !
;  368     0366  1 !       None.
;  369     0367  1 !
;  370     0368  1 ! Routine value:
;  371     0369  1 !
;  372     0370  1 !       None.
;  373     0371  1 !--
;  374     0372  1
;  375     0373  2 BEGIN                                           ! Start of REQUEST_CLM_HANDLER
;  376     0374  2
;  377     0375  2 LOCAL
;  378     0376  2         RQCB            : $ref_bblock,          ! RQCB data structure
;  379     0377  2         QCD             : $ref_bblock,          ! QCD data structure
;  380     0378  2         MCB             : $ref_bblock,          ! MCB data structure
;  381     0379  2         MSG             : $ref_bblock,          ! Pointer to user request
;  382     0380  2         FOUND           : LONG,                 ! Boolean
;  383     0381  2         SCOPE           : LONG,                 ! Scope of request
;  384     0382  2         SCOPE_LIMIT     : LONG,                 ! Loop control
;  385     0383  2         STATUS          : LONG;
;  386     0384  2
;  387     0385  2
;  388     0386  2 !
;  389     0387  2 ! Check the version number of the message.  If the message is from any other version,
;  390     0388  2 ! simply ignore it.
;  391     0389  2 !
;  392     0390  2 IF .CLM [CLM_B_DS_VERSION] NEQ CLMRQCB_K_DS_VERSION
;  393     0391  2 THEN
;  394     0392  2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLM__REQUEST mismatch');
;  395     0393  2 !
```

```
;  396     0394  2 ! Allocate an RQCB and convert the message RQCB into the new RQCB
;  397     0395  2 !
;  398     0396  2 IF NOT CLUSMSG_CONV_CLM_RQCB (.CLM, RQCB)
;  399     0397  2 THEN
;  400     0398  2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, ascid_INVALIDRQCB);
;  401     0399  2 !
;  402     0400  2 ! Find an OCD that can handle this request. The OCD is selected
;  403     0401  2 ! according to the SCOPE and UIC of the requestor.  If the SCOPE
;  404     0402  2 ! is unspecified, then look for operator coverage starting in the
;  405     0403  2 ! least privileged scope and continuing to the most privileged.
;  406     0404  2 ! If no OCD is found, then dismiss the request.
;  407     0405  2 !
;  408     0406  2 IF (.RQCB [RQCB_B_SCOPE] EQL OPC$K_UNSPEC)
;  409     0407  2 THEN
;  410     0408  2     SCOPE_LIMIT = OPC$K_SYSTEM
;  411     0409  2 ELSE
;  412     0410  2     SCOPE_LIMIT = .RQCB [RQCB_B_SCOPE];
;  413     0411  2 FOUND = FALSE;
;  414     0412  2 SCOPE = .RQCB [RQCB_B_SCOPE];
;  415     0413  2 WHILE (.SCOPE GEQ .SCOPE_LIMIT) AND (NOT .FOUND) DO
;  416     0414  3     IF NOT (FOUND = FIND_OCD (.SCOPE, .RQCB [RQCB_L_UIC], OCD))
;  417     0415  3     THEN
;  418     0416  3         SCOPE = .SCOPE - 1;
;  419     0417  2 IF NOT .FOUND
;  420     0418  2 THEN
;  421     0419  3     BEGIN
;  422     0420  3     DEALLOCATE_RQCB (.RQCB);
;  423     0421  3     RETURN;
;  424     0422  3     END;
;  425     0423  2 RQCB [RQCB_L_OCD] = .OCD;                       ! Save OCD address
;  426     0424  2 RQCB [RQCB_B_SCOPE] = .OCD [OCD_B_SCOPE];       ! Set request scope
;  427     0425  2 !
;  428     0426  2 ! Tell the world about the request, first to the log file, then to the operators.  We
;  429     0427  2 ! know that an operator was notified, otherwise the remote node would not have sent the
;  430     0428  2 ! message.
;  431     0429  2 !
;  432     0430  2 LOG_MESSAGE (.RQCB);
;  433     0431  2 NOTIFY_LISTED_OPERATORS (.RQCB);
;  434     0432  2 !
;  435     0433  2 ! At least one operator was notified of the request.  If the request expects a reply,
;  436     0434  2 ! then queue the RQCB onto the OCD for future reference.
;  437     0435  2 !
;  438     0436  2 IF .RQCB [RQCB_W_REPLYMBX] NEQ 0
;  439     0437  2 THEN
;  440     0438  3     BEGIN
;  441     0439  3     INSQUE (.RQCB, .OCD [OCD_L_RQSTFLINK]);
;  442     0440  3     OCD [OCD_W_RQSTCOUNT] = .OCD [OCD_W_RQSTCOUNT] + 1;
;  443     0441  3     $bblock [RQCB [RQCB_L_OPTIONS], OPC$V_NOBRD] = 0;   ! Clear option bits
;  444     0442  3     $bblock [RQCB [RQCB_L_OPTIONS], OPC$V_NOLOG] = 0;
;  445     0443  3     END
;  446     0444  2 ELSE
;  447     0445  2     DEALLOCATE_RQCB (.RQCB);                        ! Dellocate the RQCB
;  448     0446  2
;  449     0447  1 END;                                            ! End of REQUEST_CLM_HANDLER
```

```
                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

69 6D 20 54 53 45 55 51 45 52 5F 5F 4D 4C 43  00000 P.AAB:   .ASCII  \CLM__REQUEST mismatch\<0><0><0>                ;
                  00 00 00 68 63 74 61 6D 73  0000F                                                              ;
                                              00018 P.AAA:   .LONG   17694741                                     ;
                                              0001C          .ADDRESS P.AAB                                       ;

                                                             .EXTRN  ASCID_INVALIDRQCB

                                                             .PSECT  $CODE$,NOWRT,2

                                              00000          .ENTRY  REQUEST_CLM_HANDLER, Save R2,R3,R4           ; 0337
                                    5E  08 C2 00002          SUBL2   #8, SP                                       ;
                                    52  08 AC D0 00005          MOVL    CLM, R2                                  ; 0390
                                    02  02 A2 91 00009          CMPB    2(R2), #2                                ;
                                           06 13 0000D          BEQL    1$                                       ;
                                 0000' CF 9F 0000F          PUSHAB  P.AAA                                        ; 0392
                                           10 11 00013          BRB     2$                                       ;
                                 4004 8F BB 00015 1$:       PUSHR   #^M<R2,SP>                                   ; 0396
                          0000G  CF      02 FB 00019          CALLS   #2, CLUSMSG_CONV_CLM_RQCB                    ;
                                 0D      50 E8 0001E          BLBS    R0, 3$                                       ;
                                 0000G CF 9F 00021          PUSHAB  ASCID_INVALIDRQCB                            ; 0398
                                    04 AC DD 00025 2$:      PUSHL   BUFFER_DESC                                  ;
                          0000G  CF      02 FB 00028          CALLS   #2, DUMP_LOG_FILE                            ;
                                           04 0002D          RET                                                  ;
                                    52  6E D0 0002E 3$:      MOVL    RQCB, R2                                     ; 0406
                                    04  53 A2 91 00031          CMPB    83(R2), #4                               ;
                                           05 12 00035          BNEQ    4$                                       ;
                                    54     01 D0 00037          MOVL    #1, SCOPE_LIMIT                          ; 0408
                                           04 11 0003A          BRB     5$                                       ;
                                    54  53 A2 9A 0003C 4$:      MOVZBL  83(R2), SCOPE_LIMIT                      ; 0410
                                    50     D4 00040 5$:        CLRL    FOUND                                    ; 0411
                                    53  53 A2 9A 00042          MOVZBL  83(R2), SCOPE                            ; 0412
                                    54     53 D1 00046 6$:      CMPL    SCOPE, SCOPE_LIMIT                       ; 0413
                                           17 19 00049          BLSS    7$                                       ;
                                    17     50 E8 0004B          BLBS    FOUND, 8$                                ;
                                       04  AE 9F 0004E          PUSHAB  OCD                                      ; 0414
                                       68  A2 DD 00051          PUSHL   104(R2)                                  ;
                                           53 DD 00054          PUSHL   SCOPE                                    ;
                          0000G  CF      03 FB 00056          CALLS   #3, FIND_OCD                                 ;
                                 E8      50 E8 0005B          BLBS    FOUND, 6$                                    ;
                                           53 D7 0005E          DECL    SCOPE                                    ; 0416
                                           E4 11 00060          BRB     6$                                       ; 0414
                                    33     50 E9 00062 7$:      BLBC    FOUND, 9$                                ; 0417
                                    53  04 AE D0 00065 8$:      MOVL    OCD, R3                                  ; 0423
                                 24 A2     53 D0 00069          MOVL    R3, 36(R2)                               ;
                                 53 A2  0B  A3 90 0006D          MOVB    11(R3), 83(R2)                           ; 0424
                                           52 DD 00072          PUSHL   R2                                       ; 0430
                          0000G  CF      01 FB 00074          CALLS   #1, LOG_MESSAGE                              ;
                                           52 DD 00079          PUSHL   R2                                       ; 0431
                          0000G  CF      01 FB 0007B          CALLS   #1, NOTIFY_LISTED_OPERATORS                  ;
                                    2E  A2 B5 00080          TSTW    46(R2)                                       ; 0436
                                           13 13 00083          BEQL    9$                                       ;
                                 3C  B3    62 0E 00085          INSQUE  (R2), @60(R3)                            ; 0439
                                    50  04 AE D0 00089          MOVL    OCD, R0                                  ; 0440
                                        3A A0 B6 0008D          INCW    58(R0)                                   ;
                                    50     6E D0 00090          MOVL    RQCB, R0                                 ; 0441
```

```
                          54   A0          03  8A 00093         BICB2   #3, 84(R0)                      ; 0442
                                           04  00097         RET                                     ; 0436
                                           52  DD 00098 9$:     PUSHL   R2                              ; 0445
                        0000G  CF          01  FB 0009A         CALLS   #1, DEALLOCATE_RQCB             ;
                                           04  0009F         RET                                     ; 0447

; Routine Size:  160 bytes,    Routine Base:  $CODE$ + 0203
```

```
;  451     0448  1 GLOBAL ROUTINE REQUEST_CLM_CHECK_HANDLER (BUFFER_DESC : $ref_bblock, CLM : $ref_bblock, LEN) : NOVALUE =
;  452     0449  1
;  453     0450  1 !++
;  454     0451  1 ! Functional description:
;  455     0452  1 !
;  456     0453  1 !       This routine is the handler for all CHECK_REQUEST messages received by OPCOM from remote nodes.
;  457     0454  1 !
;  458     0455  1 !
;  459     0456  1 ! Input:
;  460     0457  1 !
;  461     0458  1 !       BUFFER_DESC -   pointer to message from remote node, including $SNDOPR header
;  462     0459  1 !       CLM -           pointer to CLMRQCB structure
;  463     0460  1 !       LEN -           length of LEN
;  464     0461  1 !
;  465     0462  1 ! Implicit Input:
;  466     0463  1 !
;  467     0464  1 !       None.
;  468     0465  1 !
;  469     0466  1 ! Output:
;  470     0467  1 !
;  471     0468  1 !       None.
;  472     0469  1 !
;  473     0470  1 ! Implict output:
;  474     0471  1 !
;  475     0472  1 !       Some accounting data will be updated
;  476     0473  1 !       to reflect the receipt of the message.
;  477     0474  1 !
;  478     0475  1 ! Side effects:
;  479     0476  1 !
;  480     0477  1 !       None.
;  481     0478  1 !
;  482     0479  1 ! Routine value:
;  483     0480  1 !
;  484     0481  1 !       None.
;  485     0482  1 !--
;  486     0483  1
;  487     0484  2 BEGIN                                           ! Start of REQUEST_CLM_CHECK_HANDLER
;  488     0485  2
;  489     0486  2 LOCAL
;  490     0487  2         RQST            : $ref_bblock,          ! RQCB data structure
;  491     0488  2         RQCB            : $ref_bblock,          ! RQCB data structure
;  492     0489  2         OCD             : $ref_bblock,          ! OCD data structure
;  493     0490  2         MCB             : $ref_bblock,          ! MCB data structure
;  494     0491  2         MSG             : $ref_bblock,          ! Pointer to user request
;  495     0492  2         RQST_COUNT      : LONG,                 ! Count of requests
;  496     0493  2         FOUND           : LONG,                 ! Boolean
;  497     0494  2         SCOPE           : LONG,                 ! Scope of request
;  498     0495  2         SCOPE_LIMIT     : LONG,                 ! Loop control
;  499     0496  2         STATUS          : LONG;
;  500     0497  2
;  501     0498  2 !
;  502     0499  2 ! Check the version number of the message.  If the message is from any other version,
;  503     0500  2 ! simply ignore it.
;  504     0501  2 !
;  505     0502  2 IF .CLM [CLM_B_DS_VERSION] NEQ CLMRQCB_K_DS_VERSION
;  506     0503  2 THEN
;  507     0504  2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, %ASCID 'CLM__CHECK_REQUEST mismatch');
```

```
;  508          0505  2 !
;  509          0506  2 ! Allocate an RQCB and convert the message RQCB into the new RQCB
;  510          0507  2 !
;  511          0508  2 IF NOT CLUSMSG_CONV_CLM_RQCB (.CLM, RQCB)
;  512          0509  2 THEN
;  513          0510  2     RETURN DUMP_LOG_FILE (.BUFFER_DESC, ascid_INVALIDRQCB);
;  514          0511  2 !
;  515          0512  2 ! Find an OCD that can handle this request. The OCD is selected according to the SCOPE and UIC of the
;  516          0513  2 ! requestor.  If the SCOPE is unspecified, then look for operator coverage starting in the least
;  517          0514  2 ! privileged scope and continuing to the most privileged.  If no OCD is found, then dismiss the request.
;  518          0515  2 !
;  519          0516  2 IF (.RQCB [RQCB_B_SCOPE] EQL OPC$K_UNSPEC)
;  520          0517  2 THEN
;  521          0518  2     SCOPE_LIMIT = OPC$K_SYSTEM
;  522          0519  2 ELSE
;  523          0520  2     SCOPE_LIMIT = .RQCB [RQCB_B_SCOPE];
;  524          0521  2 FOUND = FALSE;
;  525          0522  2 SCOPE = .RQCB [RQCB_B_SCOPE];
;  526          0523  2 WHILE (.SCOPE GEQ .SCOPE_LIMIT) AND (NOT .FOUND) DO
;  527          0524  3     IF NOT (FOUND = FIND_OCD (.SCOPE, .RQCB [RQCB_L_UIC], OCD))
;  528          0525  3     THEN
;  529          0526  4         SCOPE = .SCOPE - 1;
;  530          0527  2 IF NOT .FOUND
;  531          0528  2 THEN
;  532          0529  3     BEGIN
;  533          0530  3     DEALLOCATE_RQCB (.RQCB);
;  534          0531  3     RETURN;
;  535          0532  3     END;
;  536          0533  2 RQCB [RQCB_L_OCD] = .OCD;                       ! Save OCD address
;  537          0534  2 RQCB [RQCB_B_SCOPE] = .OCD [OCD_B_SCOPE];       ! Set request scope
;  538          0535  2 !
;  539          0536  2 ! Search through the requests queued to this OCD for the specified request.  If it is already present,
;  540          0537  2 ! then free the RQCB and return.
;  541          0538  2 !
;  542          0539  2 RQST_COUNT = .OCD [OCD_W_RQSTCOUNT];    ! Get # of requests
;  543          0540  2 RQST = .OCD [OCD_L_RQSTFLINK];          ! Get first request address
;  544          0541  2 WHILE .RQST_COUNT GTR 0 DO
;  545          0542  3     BEGIN
;  546          0543  3     IF .RQCB [RQCB_L_RQSTNUM] NEQ .RQST [RQCB_L_RQSTNUM]
;  547          0544  3     THEN
;  548          0545  4         BEGIN
;  549          0546  4         RQST_COUNT = .RQST_COUNT - 1;   ! Decrement request count
;  550          0547  4         RQST = .RQST [RQCB_L_FLINK];    ! Get address of next request RQCB
;  551          0548  4         END
;  552          0549  3     ELSE
;  553          0550  4         BEGIN
;  554          0551  4         DEALLOCATE_RQCB (.RQCB);
;  555          0552  4         RETURN;
;  556          0553  3         END;
;  557          0554  2     END;
;  558          0555  2 !
;  559          0556  2 ! Tell the world about the request, first to the log file, then to the operators.  We
;  560          0557  2 ! know that an operator was notified, otherwise the remote node would not have sent the
;  561          0558  2 ! message.
;  562          0559  2 !
;  563          0560  2 LOG_MESSAGE (.RQCB);
;  564          0561  2 NOTIFY_LISTED_OPERATORS (.RQCB);
```

```
;  565          0562  2 !
;  566          0563  2 ! Everything looks good, add it to the list
;  567          0564  2 !
;  568          0565  2 INSQUE (.RQCB, .OCD [OCD_L_RQSTFLINK]);
;  569          0566  2 OCD [OCD_W_RQSTCOUNT] = .OCD [OCD_W_RQSTCOUNT] + 1;
;  570          0567  2
;  571          0568  1 END;                                    ! End of REQUEST_CLM_CHECK_HANDLER


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

55 51 45 52 5F 4B 43 45 48 43 5F 5F 4D 4C 43  00020 P.AAD:  .ASCII  \CLM__CHECK_REQUEST mismatch\<0>                ;
      00 68 63 74 61 6D 73 69 6D 20 54 53 45  0002F                                                              ;
                                 010E001B  0003C P.AAC:  .LONG   17694747                                         ;
                                 00000000'  00040        .ADDRESS P.AAD                                           ;


                                                        .PSECT  $CODE$,NOWRT,2

                                    003C  00000         .ENTRY  REQUEST_CLM_CHECK_HANDLER, Save R2,R3,R4,R5 ; 0448
                              5E          08  C2  00002         SUBL2   #8, SP                                  ;
                              52    08    AC  D0  00005         MOVL    CLM, R2                                 ; 0502
                              02    02    A2  91  00009         CMPB    2(R2), #2                               ;
                                          06  13  0000D         BEQL    1$                                      ;
                                    0000' CF  9F  0000F         PUSHAB  P.AAC                                   ; 0504
                                          10  11  00013         BRB     2$                                      ;
                                    4004  8F  BB  00015 1$:     PUSHR   #^M<R2,SP>                              ; 0508
                        0000G     CF        02  FB  00019         CALLS   #2, CLUSMSG_CONV_CLM_RQCB               ;
                                  0D        50  E8  0001E         BLBS    R0, 3$                                  ;
                                    0000G CF  9F  00021         PUSHAB  ASCID_INVALIDRQCB                       ; 0510
                                      04  AC  DD  00025 2$:     PUSHL   BUFFER_DESC                             ;
                        0000G     CF        02  FB  00028         CALLS   #2, DUMP_LOG_FILE                       ;
                                            04  0002D         RET                                             ;
                                  53        6E  D0  0002E 3$:     MOVL    RQCB, R3                                ; 0516
                                  04    53  A3  91  00031         CMPB    83(R3), #4                              ;
                                            05  12  00035         BNEQ    4$                                      ;
                                  54        01  D0  00037         MOVL    #1, SCOPE_LIMIT                         ; 0518
                                            04  11  0003A         BRB     5$                                      ;
                                  54    53  A3  9A  0003C 4$:     MOVZBL  83(R3), SCOPE_LIMIT                     ; 0520
                                            50  D4  00040 5$:     CLRL    FOUND                                   ; 0521
                                  52    53  A3  9A  00042         MOVZBL  83(R3), SCOPE                           ; 0522
                                  54        52  D1  00046 6$:     CMPL    SCOPE, SCOPE_LIMIT                      ; 0523
                                            17  19  00049         BLSS    7$                                      ;
                                  17        50  E8  0004B         BLBS    FOUND, 8$                               ;
                                      04    AE  9F  0004E         PUSHAB  OCD                                     ; 0524
                                      68    A3  DD  00051         PUSHL   104(R3)                                 ;
                                            52  DD  00054         PUSHL   SCOPE                                   ;
                        0000G     CF        03  FB  00056         CALLS   #3, FIND_OCD                            ;
                                  E8        50  E8  0005B         BLBS    FOUND, 6$                               ;
                                            52  D7  0005E         DECL    SCOPE                                   ; 0526
                                            E4  11  00060         BRB     6$                                      ; 0524
                                  27        50  E9  00062 7$:     BLBC    FOUND, 10$                              ; 0527
                                  52    04  AE  D0  00065 8$:     MOVL    OCD, R2                                 ; 0533
                              24  A3        52  D0  00069         MOVL    R2, 36(R3)                              ;
                              53  A3    0B  A2  90  0006D         MOVB    11(R2), 83(R3)                          ; 0534
```

```
                        55        3A    A2  3C 00072          MOVZWL  58(R2), RQST_COUNT                ; 0539
                        54        3C    A2  D0 00076          MOVL    60(R2), RQST                     ; 0540
                                        55  D5 0007A 9$:       TSTL    RQST_COUNT                       ; 0541
                                        16  15 0007C          BLEQ    11$                              ;
                      70    A4        70  A3  D1 0007E          CMPL    112(R3), 112(RQST)               ; 0543
                                        07  13 00083          BEQL    10$                              ;
                                        55  D7 00085          DECL    RQST_COUNT                       ; 0546
                        54              64  D0 00087          MOVL    (RQST), RQST                     ; 0547
                                        EE  11 0008A          BRB     9$                               ; 0543
                                        53  DD 0008C 10$:      PUSHL   R3                               ; 0551
                0000G   CF              01  FB 0008E          CALLS   #1, DEALLOCATE_RQCB              ;
                                            04 00093          RET                                      ; 0550
                                        53  DD 00094 11$:      PUSHL   R3                               ; 0560
                0000G   CF              01  FB 00096          CALLS   #1, LOG_MESSAGE                  ;
                                        53  DD 0009B          PUSHL   R3                               ; 0561
                0000G   CF              01  FB 0009D          CALLS   #1, NOTIFY_LISTED_OPERATORS      ;
                  3C    B2              63  0E 000A2          INSQUE  (R3), @60(R2)                    ; 0565
                        50        04    AE  D0 000A6          MOVL    04(R3), R0                       ; 0566
                                  3A    A0  B6 000AA          INCW    58(R0)                           ;
                                            04 000AD          RET                                      ; 0568

; Routine Size:  174 bytes,    Routine Base:  $CODE$ + 02A3
```

```
;  573          0569  1 END                          ! End of OPCOMRQST
;  574          0570  0 ELUDOM
```

```
;                                PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         849  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          68  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                             Library Statistics

;                                    --------  Symbols  --------      Pages       Processing
;       File                         Total     Loaded    Percent     Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1      18619         13          0       1000        00:01.9
;  _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1   633         32          5         43        00:00.9
```

```
;                                  COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:OPCOMRQST/OBJ=OBJ$:OPCOMRQST MSRC$:OPCOMRQST/UPDATE=(ENH$:OPCOMRQST)

; Size:          849 code + 68 data bytes
; Run Time:        00:20.1
; Elapsed Time:    00:54.7
; Lines/CPU Min:    1703
; Lexemes/CPU-Min: 16488
; Memory Used:  183 pages
; Compilation Complete
```

LOGFILE
LIS

OPCOMDEF
LIS

OPCOMDATA
LIS

OPCOMINI
LIS

OPCOMUTIL
LIS

OPCOMLIB
LIS

OPCDEFTMP
LIS

OPCOMMAIN
LIS

OPCOMOLD
LIS

OPCOMRPLY
LIS

OPCCRASH
LIS

OPERUTIL
LIS

OPCOMRQST
LIS